EMAS 2900 SUPERVISOR NOTE

No: 1 (revised)
Date: 16/4/81

PON Mechanism for Director

Each process has four service numbers associated with it:

process+N1, process+N2, process+N3, process+N4

Currently, N1=64 (X'40'), and N2, N3 and N4 vary with the configuration. Service 'process+N1' refers to the Local Controller and should not be used by anyone but P.D.S.!!! N2, N3 and N4 may be found from COM_SYNC1DEST, COM_SYNC2DEST and COM_ASYNCDEST respectively (see Supervisor Note 15).

Services 'process+N2' and 'process+N3' refer to the Director. They are identical in the way they are handled by the Kernel and the Local Controller but the conventional use is intended to be that 'process+N2' should be the normal service for passing messages to the Director and that 'process+N3' should be used when a specific message (usually a reply) is being awaited. 'process+N4' also refers to the Director but is unlike the other two in that messages received on this service will be dealt with immediately, rather than waiting for a correspnding Director POFF. The normal effect is to enter Director's SIGNAL state.

OUTs 5 and 6 refer to service 'process+N2', OUTs 7 and 8 refer to service 'process+N3', and OUTs 9 and 10 refer to service 'process+N4'. OUTs 5, 7 and 9 imply that the process is to be suspended until a suitable message is available, and OUTs 6, 8 and 10 imply that the process wishes to continue processing after the PON.

All the OUTs expect the service to be PONned to be in DIROUTP. If DIROUTP_DEST>>16 is zero then no PON is done. This gives a pure 'suspend' for OUTs 5, 7 and 9. For OUTs 6, 8 and 10 with a DEST of zero it gives a TOFF. That is, a POFF(DIROUTP) is done on the appropriate service number. If no parameters are waiting, DIROUTP_DEST is set to zero. The source service number is inserted by the Local Controller, leaving the source activity unaltered. The inhibit/uninhibit mechanism operates so that 'process+N2' for OUT 5, and 'process+N3' for OUT 7, are uninhibited only when the process is suspended (the alternate service remains inhibited). The 'process+N4' service number is always uninihibited whenever the process is suspended.

Summary of effects of these OUTs

	Message type	Effect: DEST non-zero	Effect: DEST zero
OUT 5	SYNC1DEST	PON+POFF	POFF
OUT 6		PON (and continue)	TOFF
OUT 7	SYNC2DEST	PON+POFF	POFF
OUT 8		PON (and continue)	TOFF
OUT 9	ASYNCDEST	PON	-
OUT10		PON (and continue)	TOFF

It is also possible, through the use of OUTs 5-10, to send a message asynchronously to a process identified by means of its user name. This is known as the 'relay' facility.

To use it, DIROUTP_DEST should be set to X'FFFF' and the user name of the destination process should be loaded, left justified in ACC (128 bits, i.e. ACS=3), as a string(6). The Local Controller searches for this user name and replaces the X'FFFF' with the appropriate service number before doing the PON. The destination and source activities are unchanged, as standard. The source service numbers are also set as standard.

If the user name cannot be found, i.e. nothing logged on under that name, DIROUTP_DEST is set to zero and control is returned immediately to the Director. Otherwise, DIROUTP_DEST will be non-zero on return.

The relay facility allows messages to be PONned to any of the three local process service numbers. An additional parameter is required which takes the values 1, 2 or 3 to specify the local service number, the alternative local service number or the asynchronous service number, respectively. These are the analogue of the reply service numbers generated by OUT 5 and OUT 6, OUT 7 and OUT 8 or OUT 9 and OUT 10 respectively. This parameter is expected in the rightmost word of the accumulator. For example:

6	E	C	S	C	0	4	inc no		3

To enable different incarnations of a process to be distinguished, the incarnation number, in binary, can be appended to the string (as shown).

PON and Wait

An OUT 11 has been introduced which allows a message to be PONned and the process to remain in store until a reply arrives. Needless to say, in view of the possible blockage, this will need to be used very circumspectly; in particular for things like magnetic tape transfer, which one expects to take place quickly.

Due to the way this has to be implemented (using the Local Controller service number as the reply destination) there is no possibility of specifying any of the alternative local service numbers as a reply service number. The implications of this are clear, I hope. Also, because of the possibility of blockage and deadly embrace, only Kernel services can be PONned in this way and not messages to processes. The 'relay' facility is therefore not relevant here.

Delayed relay service number 63

Any standard parameter record ponned to service 63 activity N will be relayed, after a delay of N seconds, to the DEST/DACT in P_P6. P_SRCE, P_P1, P_P2, P_P3, P_P4 and P_P5 are left unchanged. N=0 is acceptable, if unwise.

These facilities are available from SUP26H onwards.

D.J. Rees
P.D. Stephens
J.K. Yarwood

EMAS 2900 SUPERVISOR NOTE

No: 2 (revised)
Date: 21/4/81

Data Passed to Director on Initial Entry

When Director is first activated, the B register contains the address of the Director / Local Controller parameter-passing record DIROUTP. This register is preserved by the Dirloader and passed onto Director as a recordname parameter.

DIROUTP contains useful information, including the address of the signal record SIGOUTP, which has further initialisation data.

Both records have the standard format:

recordformat PARMF (integer DEST, SRCE, P1, P2, P3, P4, P5, P6)

They are initialised as follows:

DIROUTP

_DEST = Supervisor identifier, as string(3), e.g. "25F"

_SRCE = epage size <<16 ! max disc block size

_P1 = process number

_P2 and _P3 = username, as string(6), e.g. "MANAGR", followed by a byte giving the incarnation number.

_P4 = address of SIGOUTP

_P5 = address of System Call Index Table

_P6 = DACT to be used for terminal INT messages

SIGOUTP

_DEST = LST LEN, i.e. number of segments in Local Segment Table

_SRCE = address of Secondary Segment Table

_P1 = index of last Claimed Block Table entry

_P2 = address of Claimed Block Table

_P3 = address of accounting record

_P4 = address of instruction counter revolutions count

_P5 = address of Input/Output status record

_P6 = not used

P.D. Stephens

EMAS 2900 SUPERVISOR NOTE

No: 3 (revised)
Date: 21/4/81

Tables and Formats Describing the VM

The VM is described by a series of tables indexed from the Secondary Segment Table, which parallels the hardware Segment Table.

The formats of the tables are as follows:

SST byteinteger (0:segment table length-1)

CBT (integer DA, halfinteger AMTX, byteinteger TAGS, TAGS2)

DA		disc address of (start of) block
AMTX		index to Active Memory Table for block (valid only when active bit set in TAGS)
TAGS	bit 7	new, i.e. all pages in this block are initially zero
	bit 6	advise disc only (i.e. drum not to be used)
	bit 5	active, i.e. AMT entry allocated
	bits 4->0	limit (in epages) of block. Range 0-31, i.e. blocks up to 128 Kbytes
TAGS2	bit 7	continuation, i.e. this is not the first block in the file
	bit 6	advisory forward sequential - Supervisor may page the block assuming (repeated) forward sequential access
	bits 5->0	unused

AMTHASH ownhalfintegerarray (0:511)

list heads of chains of AMT entries whose disc addresses hash to the same value

AMT (integer DA, halfinteger DDP, LINK, USERS, byteinteger LEN, OUTS)

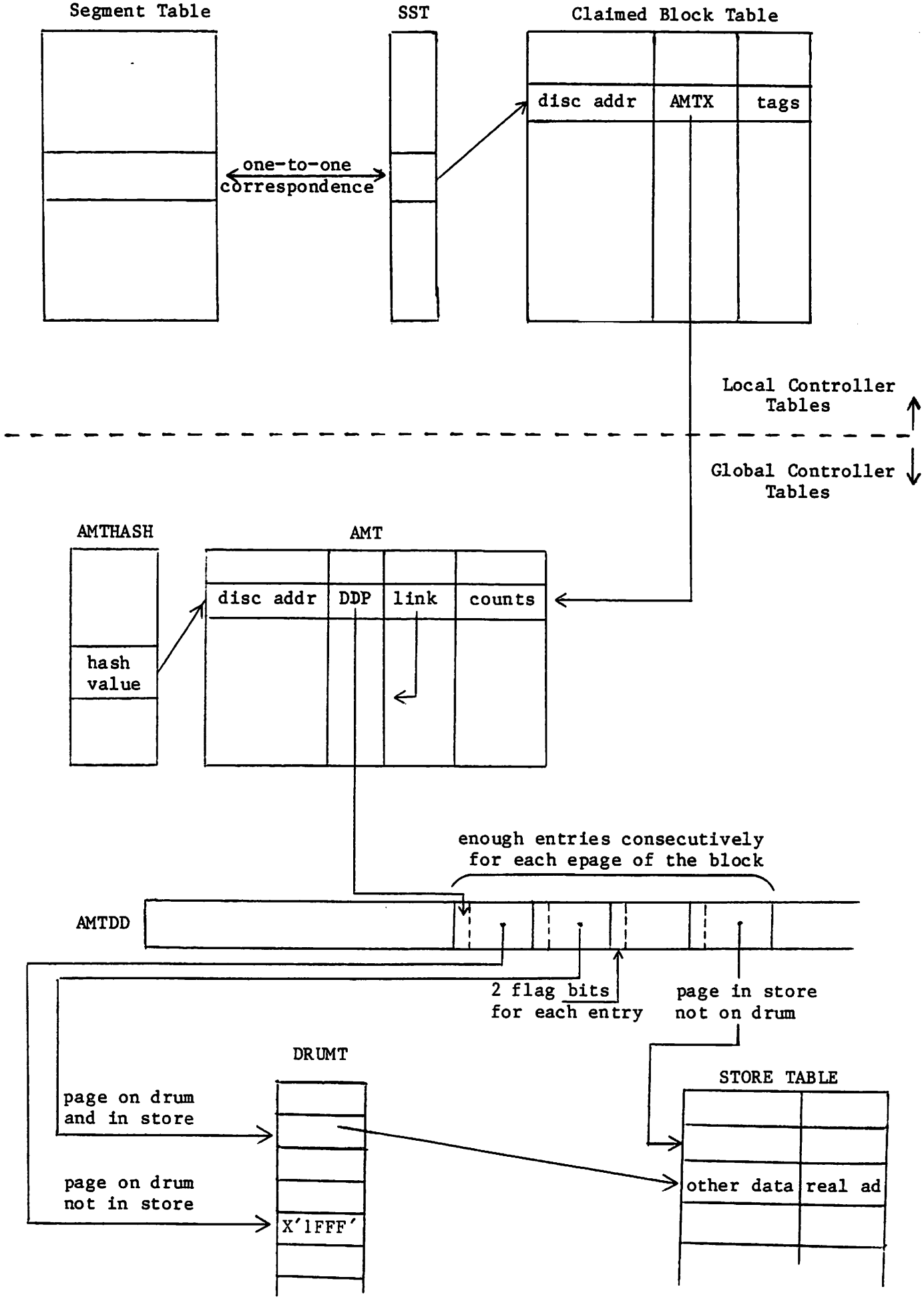
DA	disc address of block
DDP	index into AMTDD array for the start of block
LINK	link to other AMTs having the same disc address hash key
USERS	number of users of this block plus users of each page in this block
LEN	length of block
OUTS	number of page-outs in progress for this block

AMTDD halfintegerarray (0:extensible in epage units as required)

15	"new" epage
bit 14	0 Store Table Index follows
	1 Drum Table Index follows
bits 13->0	Store or Drum Table Index (X'1FFF' indicates no store epage allocated)

DRUMT halfintegerarray (0:drum size -1)

bits 13->0 Store Table Index (X'1FFF' if unallocated)


Segment Table
SST
Claimed Block Table
disc addr
AMTX
tags
one-to-one
correspondence
Local Controller
Tables
Global Controller
Tables
AMTHASH
AMT
disc addr
DDP
link
counts
hash
value
enough entries consecutively
for each epage of the block
AMTDD
2 flag bits
for each entry
page in store
not on drum
DRUMT
STORE TABLE
page on drum
and in store
page on drum
not in store
X'1FFF'
other data
real ad

Note the one-to-one correspondence between DRUMT and logical drum epages such that the Drum Table Index doubles as the logical drum address. This precludes any "alternative track allocations" but this is unimportant: if the drum page is defective the information can be fetched from disc.

```
STORE    (byteinteger FLAGS, USERS, halfinteger, LINK, c
            BLINK, FLINK, integer REALAD)
```

(one entry per usable epage)

FLAGS	bits 7->6	11: disc write in progress 10: disc read in progress 00: no disc transfer in progress
	bits 5->4	11: drum write in progress 10: drum read in progress 00: no drum transfer in progress
	bit 3	"written to" since leaving drum or disc
	bit 2	0: page-out to disc only 1: pageout to disc and drum
	bit 1	make "new" upon page-out
	bit 0	recapturable (when on free list)
USERS	number of users of this page	
LINK	link to page-in-transit list (when a read is in progress only), or (when on free list and recapturable):	
	bit 15	0: AMTDD Index follows 1: DRUMT Index follows
	bit 14->0	AMTDD or DRUMT Index
BLINK	back link to previous entry on free list; if page is in use then this is EPX, the page number in the block defined by AMTX	
FLINK	forward link to next entry on free list; if page is in use then this is AMTX	
REALAD	real store address of this epage	

Page-in-transit Tables

These are lists of parameter cells containing reply parameters for all those processes awaiting the epage.

AMT and AMTDD

They are each mapped into their own (public) segments (21 and 22 respectively), so that they can be extended as required by appending an epage and adding the new space to the appropriate free lists.

Free lists

The tables in general have free lists linking unused entries:

CBT — Free entries are identified by being zero.

AMT — A pushdown list of free entries linked via the LINK field. The list head is AMTASL, in routine ACTIVE MEM. This table can be extended as necessary.

AMTDD — Separate pushdown free lists for each possible size of entry, i.e. 1-> MAXBLOCK epages. The list heads are kept in array DDASL(1:MAXBLOCK), in routine ACTIVE MEM. If a free list is empty a three-step procedure is used:

1) An attempt is made to fragment a larger entry.

2) All free lists are garbage-collected to recover any fragmentation losses.

3) The AMTDD is extended by adding an epage.

DRUMT — A pushdown list is maintained (by routine PAGETURN). The listhead is DDTASL, a Global Controller variable. A special arrangement is made for returning page-frames on which transfers have failed.

STORE — A complex doubly linked list (forward links in FLINK, back links in BLINK) is maintained to allow "recaptured" pages to be removed from the middle. Global heads FSTASL and BSTASL are used for removing and adding pages respectively. This arrangement maximises the chance of recapturing pages. Non-recapturable epages are added to the head of the list.

As well as the above, each Local Controller maintains an extract of the information for efficiency reasons. This information is held for a maximum of 32 distinct active segments. The figure of 32 (which is ample; System 4 used 24 much smaller segments) is so that bit arrays (0:31) can be kept in an integer.

AS — longintegerarray (0:31)

Contains a bit for each active epage in each active segment. This array is indexed from the primary segment table using bits 26-31 (marked as "usable by software" in PLI). Note that this assumes a maximum of 64 epages/segment, i.e. a minimum epage size of 4.

ASEG — byteintegerarray (0:31)

Contains the segment number of the active segment as a back pointer to SST when needed.

ASWAP — integer masquerading as bitarray (0:31)

Marks which entries in AS actually have an active page (i.e. have been referenced) in the current residence period.

ASWIP integer masquerading as bitarray (0:31)

Marks which entries in AS are used in which no page has yet been referenced in the current residence period.

ASFREE integer masquerading as bitarray (0:31)

Marks unused entries in AS. Note that ASFREE is always equivalent to (ASWAP!ASWIP)!!(-1).

OLDASWIPS integerarray (0: MAXRESIDENCES)

Contains ASWIP at the end of the previous MAXRESIDENCES residence periods and is used to de-activate segments as they move out of the working set.

These tables are used extensively in kernel operations such as paging and strobing. They are also used in the following non-kernel operations:

a) Connection

This operation consists of Director entering the block chains into the CBT for the disc blocks of the file being connected. This is followed by setting the SST entry, and the APF and length field for the segment(s) involved.

b) Activation

This is done on a block-by-block basis and involves obtaining AMT and AMTDD space and setting the pointers. Before the AMT space is obtained a search of the relevant hash chain is made. It is at this point that shared blocks are identified.

c) De-activation

De-activation is normally done on a segment basis, when examination of OLDASWIPS shows that the segment has been active for a period. Forced de-activation is also performed when there are 32 active segments, or at the Director's request before disconnection.

There are two parts to de-activation:

Step 1: Decrement the AMT_USER fields. Destroy CBT_AMX links and reset the active bit in CBT_TAGS.

Step 2: When AMT_USER = AMT_OUTS = 0, de-allocate AMT, AMTDD and any drum space.

Step 1 is done immediately. For unshared, unused segments Step 2 follows immediately. For used blocks a delay is required while transfers complete. For shared blocks, de-activation will not complete until the last user disconnects the shared file.

d) Disconnection
e) Disconnect and Destroy

Disconnection can only be carried out on blocks marked "inactivate" in the CBT; Director must check this. Otherwise, disconnection is the inverse of connection.

A block cannot be destroyed until both steps of de-activation have been completed - otherwise corruption of the file system would be possible, by AMT registering invalid "sharing" between the old and any possible new use of the disc block. A special Local Controller service (OUT 17) is provided so that Director can wait for transfers to complete.

P.D. Stephens

EMAS 2900 SUPERVISOR NOTE

No: 4 (revised)
Date: 21/4/81

Local Stacks

To accommodate extra local stacks three new OUTs are introduced: 12, 13 and 14. Local stacks will now be numbered 1, 2, 3, etc.

At present,

Stack number 1 is the normal Director/User stack, segment 4

Stack number 2 is the signal stack, segment 6

The nomination (see below) of stacks 1 and 2 is pre-defined. At present, one more stack is available for nomination (number 3), but this can be increased.

OUT 12 : Nominate stack

DIROUTP_P1 = stack number

DIROUTP_P2 = segment number

P1 must be within bounds (1:3)
Stack number P1 must not already be nominated
P2 must be within bounds (4:max segment)
P2 must be an even number
Segment P2+1 must not be connected

OUT 13 : Denominate stack

DIROUTP_P1 = stack number

P1 must be within bounds (1:3)
Stack number P1 must be nominated
Stack number P1 must not be the current stack

OUT 14 : Swop stack

DIROUTP_P1 = new stack number

P1 must be within bounds (1:3)
Stack number P1 must be nominated
Stack number P1 must not be the current stack
The new SSN+1 context is checked to ensure that:
* LNB and SF stack segments agree
* PSR_ACS is not zero

D.J. Rees

EMAS 2900 SUPERVISOR NOTE

No: 5
Date: 8/6/78

externalroutine GPC(recordname P) - Specification

1. Conventions

The record spec for the parameter is

recordformat PARMF (integer DEST, SRCE, P1, P2, P3, P4, P5, P6)

The DEST and SRCE fields should be set up in the message to GPC as follows:

DEST considered as two halfwords DSNO and DACT, must have DSNO = GPC routine DSNO (picked up from an initialisation call to the originator), and DACT as appropriate

SRCE considered as two halfwords SSNO and SACT, must have
SSNO = SNO of originating routine, and
SACT = DACT-for-reply.

In addition, if GPC is being called or if a reply is not required, the top bit should be set in SRCE. For DACT= 4, 5 or 10 when GPC is called, the reply is the contents of the record P on return. If the request is being ponned to GPC, the top bit should not be set, and GPC will pon the reply.

2. Device entries

Each device on a GPC is allocated an area called a DEVICE ENTRY at grope time. The format of this area is given below. Fields marked * are intended to be used (read) by the calling routine; otherwise fields are either spare or for use by the GPC routine. The address of the DEVICE ENTRY for a given device is given by the GPC routine in P_P3 each time a message or reply is given to the relevant service routine, viz. in replies to ALLOCATE, DE-ALLOCATE, or EXECUTE CHAIN, in normal and abnormal TERMINATION responses, and in ATTENTION, RI and PCI responses. The format of device entry follows (all integers):

	SER	
	GPTSM	GPC-no, port, trunk, stream, mechanism
*	PROPADDR	pointer to property codes obtained at grope
	TICKS SINCE	for the time-out
*	CAA	comms area address
	GRCB AD	address of GPC's RCB
	LBA	
	ALA	
	STATE	

```
* RESP0          Response words from the most recent "user" request.
* RESP1

  SENSE1,SENSE2,
  SENSE3,SENSE4

  REPSNO

  BASE

* ID

  DLVN

* MNEMONIC       device mnemonic
```

Above this line, the format is common with that of externalroutine DISC.

```
  ENTSIZE        size of this whole entry in bytes

  PAW

  USAWO

  URCB AD        pointer to user's RCB (most recent request)

* SENSDAT AD     pointer to sense data obtained by routine GPC at most
                 recent abnormal termination for the device.

* LOGMASK        used to control GPC's logging of abnormal terminations
                 (may be altered by the relevant device service routine,
                 provided it has suitable acess to the segment). The
                 bottom byte of this word is considered as a mask for the
                 device secondary status. Following an abnormal
                 termination, if the AND of the mask and the secondary
                 status byte is non-zero, the device entry is dumped by
                 GPC on the main log.

* TRTAB AD       address of a translate table to turn characters which do
                 not exist in the repertoire into "delete" (EBCDIC 7) for
                 LP devices. (Translating with this table will have no
                 effect if CARTRIDGE is set to ZERO on the LP at grope
                 time.)

* UA SIZE        the size of the "user area" in bytes for devices to be
                 driven by ADAPTORS. These values are constants in the
                 GPC GROPE routine.

* UA AD          the address of the user area.

  TIMEOUT
```

The IMP recordformat statement defining this entry is available in file DIRARC.CFILE_ENTFORM on the Edinburgh EMAS 2900 system.

3. Initialisation

The first call of GPC must be the initialisation call (DACT=2), P1 = address of pre-prepared configuration table. DEST must contain the SNO for the GPC routine. GPC calls the following routines with a DACT of 4 for initialisation:

TAPE, OPER

4. Inputs to GPC

Apart from the initialisation call, the following DACT values are valid:

3	Interrupt from controller	P1= port+trunk (rightmost 2 quartets)
4	Allocate device	P1= device mnemonic P2= 0 CALL the service routine on interrupts. P2= 1 PON responses to the service routine.
5	De-allocate device	P1= device mnemonic
10	Execute chain	P1= RCB address P2= sno for device returned from Allocate P3= (PAW function)<<4 ! SAW flags P4= an IDENT field to be returned in P_P6 of TERMINATION and ATTENTION responses P5= 0 Normally P6= 0 Normally

Note that if P5 and P6 are not both zero, they will be taken to contain the Local Segment Table limit and base, and an ACR value, which will be written into the caller's RCB as pointed to by P_P1.

(This provides a mechanism for a privileged paged process to make direct requests on the GPC routine by using the DOUT16 Director procedure (Director Note no. 11), which ensures that a specified area of (local) virtual store resides in main store until a response is available.)

5. Responses to callers' requests

P1 = 0 means successful; otherwise:

4	Allocate 5 De-allocate	P1= 1 mnemonic invalid/not found P1= 2 already (or not) allocated

For a successful Allocate, P2 = identifier for device, to be supplied in P2 of Execute chain requests, P3 = address of the DEVICE ENTRY for the device.

10	Execute chain	P1= 1 device not found or not allocated to this service routine P1= 2 slot busy P3= address of the DEVICE ENTRY for the device

6. Action on stream response

The GPC routine response to an interrupt is to kick the appropriate service routine, which is called or ponned according to the P2 setting in the Allocate request. (Note that in the case of chains which terminate rapidly, for example faulty chains, the termination may be received before the GPC routine has returned to the caller, or ponned its acceptance of the request. In this case if responses to Execute chain requests are to cause a call of the service routine, this call would be recursive, and the replies may be received in the 'wrong' order.)

In the activation of the service routine, the record P is set as follows:

(a) For ATTENTION responses

DACT = 3

P1 = stream response word RESP0, byte 0 overwritten with device identifier as returned by Allocate. (Byte 2 is the attention byte. The RESP0 value is not written into the RESP0 field in the device entry for ATTENTION responses.)

P3 = address of the DEVICE ENTRY for the device.

(b) For NORMAL TERMINATION responses

DACT = 2

P1 = stream response word RESP0, BYTE 0 overwritten with device identifier as returned by Allocate. (Byte 2 is device primary status.)

P2 = stream response word RESP1.

P3 = address of the DEVICE ENTRY for the device.

The stream response words RESP0, RESP1 are also available in the corresponding fields in the DEVICE ENTRY.

(c) For ABNORMAL TERMINATION responses

DACT = 5

P1,P2 as for normal termination responses

P3 = address of the DEVICE ENTRY for the device

P4 = response analysis flags from the response to the SENSE command issued by GPC on behalf of the service routine. If this (right-most) byte does not indicate successful termination, then the sense information is suspect.

Status information for the device is pointed to by the field SENSAD in the DEVICE ENTRY. It is laid out as follows: secondary status followed by tertiary status (as many bytes as appropriate to the device), then primary status.

The stream response words RESP0, RESP1 are also available in the corresponding fields in the DEVICE ENTRY.

If a chain has "timed-out", due to having taken longer than a value set at grope-time for the relevant device-type, then a special abnormal termination response, in which RESP1=-1, is sent to the service routine.

(d) For PCI responses

DACT = 6

P1 = stream response word RESP0, byte 0 overwritten with the device identifier as returned by Allocate.

P3 = address of the DEVICE ENTRY for the device.

(e) For RI responses

DACT = 7

P1 = stream response word RESP0, byte 0 overwritten with device identifier as returned by Allocate.

P3 = address of the DEVICE ENTRY for the device.

J.K. Yarwood

No: 7
Date: 15/12/77

Elapsed Interval Timer

```
externalroutine ELAPSED INT (recordname P)

recordformat PARMF (integer DEST, SRCE, P1, P2, P3, P4, P5, P6)
```

P_DEST X'A0000' + activity

activity 1 - Q/UNQ nominated routine for PON every n seconds.
2 - Q nominated routine for PON in n seconds.

P_P1 service number and activity of nominated routine.

P_P2 either: number of seconds to elapse before PON(0<P_P2<X8000)
or: if activity=1 and P_P2=-1 then unqueue nominated routine.

A reply will be PONed unless P_SRCE <= 0

where P_P1 = 0 if request successful
P_P1 ≠ 0 if routine already Q'ed/UNQ'ed or P_P2 invalid.

John Maddock

EMAS 2900 SUPERVISOR NOTE

No: 8
Date: 25/1/78

Delayed PON Facility

Although the ELAPSED INT facility allows service calls to be delayed, it only allows an activity number to be specified and not a whole parameter record. A new 'DPON' routine does allow this:

externalroutinespec DPON (recordname P, integer DELAY)

The effect is to copy the parameter record P into the normal PON/POFF mechanism parameter space but not to queue it. ELAPSED INT is called to PON a service ('DPONPUTONQ', service 12) in DELAY seconds, which then does put the parameter record on the appropriate queue, thus getting the normal PON effect.

The facility is available from SUP 16D.

D.J. Rees

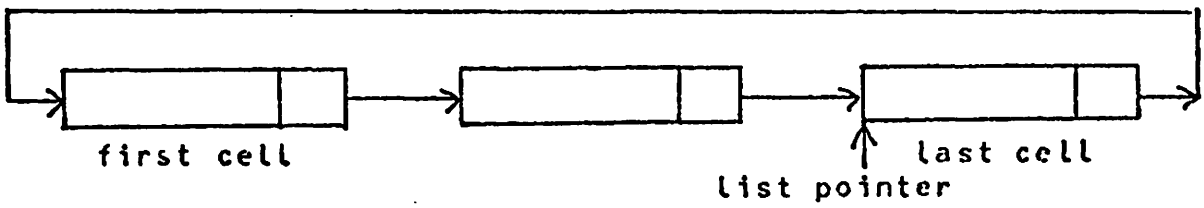
Parameter Passing Data Structures

```
%RECORDFORMAT PARMF(%INTEGER DEST,SRCE,P1,P2,P3,P4,P5,P6)
%EXTERNALRECORDARRAY PARM(1:parmn)(PARMF)
%EXTERNALINTEGERARRAY PARML(1:parmn)=2,3,4,.....
  ....,parmn,1
%EXTERNALINTEGER PARMASL=parmn

%RECORDFORMAT SERVF(%INTEGER P,L)
%EXTERNALRECORDARRAY SERVA(1:maxserv)
%EXTERNALINTEGER KERNELG,RUNQ1,RUNQ2=0
```

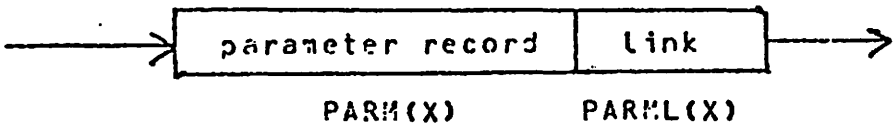
[Currently, parmn=256, maxserv=320]

All lists in this scheme are circular i.e. the link in the last cell points to the first cell. The list pointer always points to the last cell.

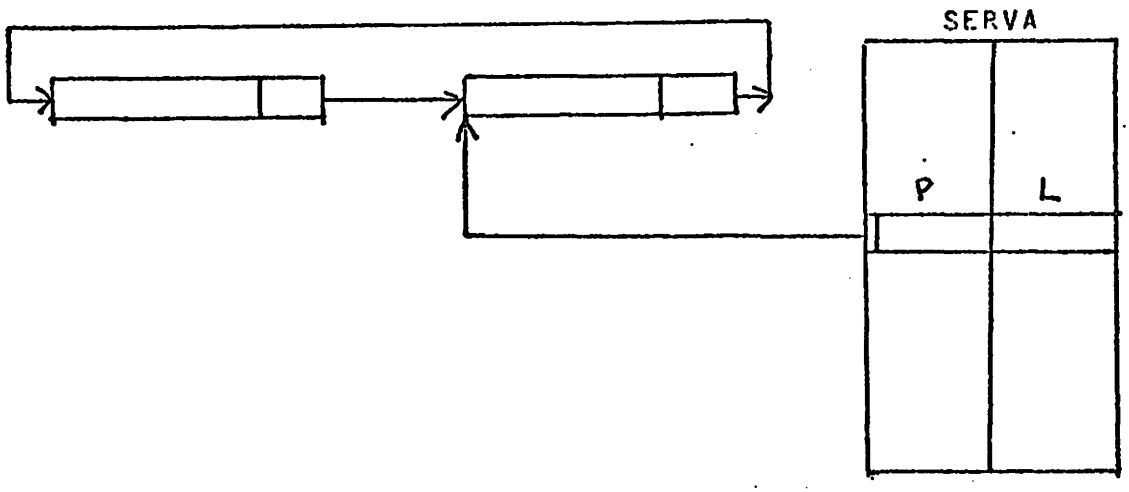


If the list contains only one cell then its link points to itself. If the list is empty the list pointer is zero.

There is a separate list for each service number which contains parameter records destined for that service. Each cell on the list consists of a pair from the parallel arrays "PARM" and "PARML".



The list pointers to these lists are held in the "SERVA" record array which is indexed by service number. Spare cells of this form are linked into an available space list with the list pointer "PARMASL".



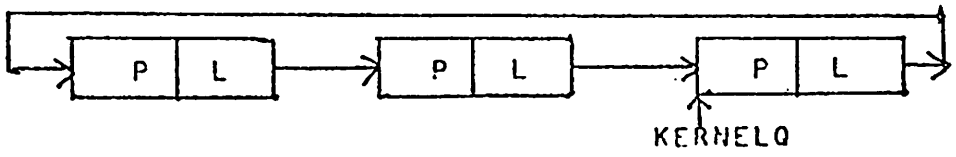
The list pointers occupy the bottom 31 bits of the record sub-element "P". The top bit i.e. the sign bit, is used to indicate the inhibit status of that service:

0 : uninhibited i.e. P>=0
1 : inhibited i.e. P<0

The "L" sub-element is used as a link for three lists which constitute a three priority-level CPU queue. The three lists, in order of priority, have pointers:

1. "KERNELQ"
2. "RUNQ1"
3. "RUNQ2"

For example, the queue of kernel services:



The "L" link values are the service numbers themselves.

D. J. Reed
17/2/78

EMAS2900 SUPERVISOR NOTE

No: 10
Date: 22/3/78

EMAS 2900: Sizes of resident code modules

1. Description of modules

Mnemonic	Description
GC	Global controller: stack-switching interrupt handler, real-time clock, main store and drum allocation low-level scheduling, process creation, hardware segment and page table maintenance, low-level resource-use monitoring.
LC	Local controller (an incarnation for each process): non-stack-switching interrupt handling (including virtual store interrupts and program errors), asynchronous message-handling for the process, timeslicing, communications message handling, process behaviour monitoring.
LC(Comms)	Communications local controller, plus LP and FE Adaptors.
MSG-SWITCH	System message switch-gear.
COMMAREA	System communication area module.
SFC	SFC (fixed-head disc) handler: queuing and execution of page-transfer requests.
DFC	DFC (removable disc) handler: queuing and execution of page-transfer and "private chain" requests, disc labelling and formatting.
GPC	GPC (tape and slow device) handler: queuing and execution of tape and slow device transfer requests (PRINT, OPER, TAPE and LINK modules formulate the requests).
OPER	OPER console handler.
PRINT	LP handler: for Supervisor log messages (these may alternatively be written to files on disc).
TAPE	Tape handler for Supervisor and paged processes.
DIAG	IMP language run-time diagnostic package for resident code.
STACK	Run-time processor stack

2. Code sizes (bytes)

Module	Code size		Work area	
	Decimal	Hexadecimal	Decimal	Hexadecimal
GC + LC	38528	X9680	6504	X1968
LC(Comms)	16312	X3FB8	7600	X1DB0
HSG-SWITCH	7160	X1BF8	12624	X3150
COMMAREA	9376	X24A0	632	X278
SFC	5856	X16E0	6936	X1B18
DFC	12184	X2F98	1024	X400
GPC	5960	X1748	592	X250
OPER	5864	X16E8	1464	X5B3
PRINT	10560	X2940	704	X2C0
TAPE	12280	X2FF8	1104	X450
DIAG	4824	X12D8	248	XF8
STACK	-	-	16384	X4000
Totals:	128904	X1F788	55816	XDA08

Total resident code + work area: 184,720 X2D190

J.K. Yarwood

2950: %EXTERNALROUTINE DCU (%RECORDNAME P) - Specification

1. Conventions

The record spec for the parameter is

recordformat PARMF (integer DEST, SRCE, P1, P2, P3, P4, P5, P6)

The DEST and SRCE fields should be set up in the message to DCU as follows:

DEST considered as two halfwords DSNO and DACT, must have DSNO = DCU routine DSNO (picked up from an initialisation call to the originator), and DACT as appropriate

SRCE considered as two halfwords SSNO and SACT, must have
SSNO = SNO of originating routine, and
SACT = DACT-for-reply.

In addition, if DCU is being called or if a reply is not required, the top bit should be set in SRCE. For DACT= 4, 5 or 10 when DCU is called, the reply is the contents of the record P on return. If the request is being ponned to DCU, the top bit should not be set, and DCU will pon the reply.

2. Device entries

Each device on a DCU is allocated an area called a DEVICE ENTRY at grope time. The format of this area is given below. Fields marked * are intended to be used (read) by the calling routine; otherwise fields are either spare or for use by the DCU routine. The address of the DEVICE ENTRY for a given device is given by the DCU routine in P_P3 each time a message or reply is given to the relevant service routine, viz. in replies to ALLOCATE, DE-ALLOCATE, or EXECUTE CHAIN, in normal and abnormal TERMINATION responses, and in ATTENTION and PCI responses. The format of device entry follows (all integers):

	SER	
	PTSM	DCU-no, stream, mechanism
*	PROPADDR	pointer to property codes obtained at grope
	SECS SINCE	for the time-out
*	CAA	comms area address (see below)
	GRCB AD	address of Transfer Control Block (TCB) used by DCU code
	LAST ATTN	last attention byte obtained from device
	ALA	not used
	STATE	

RESP0
RESP1

SENSE1,SENSE2, SENSE3,SENSE4 — Software will read Sense data into the postamble of the user's TCB, if there is one, or into these four words otherwise. This facility may be withdrawn when postambles are handled properly by the DCU. For the time being, the sense data should be found by using the SENSDAT AD field (see below).

REPSNO

BASE

* ID — Copy of P4 from parameter in last 'execute chain' command

DLVN

* MNEMONIC — device mnemonic

Above this line, the format is common with that of externalroutine DISC.

ENTSIZE — size of this whole entry in bytes

PAW

USAW0

URCB AD — pointer to user's first TCB (most recent request)

* SENSDAT AD — pointer to sense data obtained by routine DCU at most recent abnormal termination for the device.

* LOGMASK

* TRTAB AD — address of a translate table to turn characters which do not exist in the repertoire into "delete" (EBCDIC 7) for LP devices. (Translating with this table will have no effect if CARTRIDGE is set to ZERO on the LP at grope time.)

* UA SIZE — the size of the "user area" in bytes. These values are constants in the DCU GROPE routine. TCBs for the device must be held in this area.

* UA AD — the address of the user area.

TIMEOUT

PROPS0

PROPS1

The IMP recordformat statement defining this entry is available in file DIRARC.CFILE_DCUENTF on the Edinburgh EMAS 2900 system.

3. Initialisation

The first call of DCU must be the initialisation call (DACT=2), P1 = address of pre-prepared configuration table. DEST must contain the SNO for the DCU routine. DCU calls the following routines with a DACT of 4 for initialisation:

TAPE, OPER, MK 1 FE ADAPTOR

and DCU also PONs a request to the interval timer to get itself kicked every few seconds.

4. Inputs to DCU

Apart from the initialisation call, the following DACT values are valid:

3	Interrupt from controller	P1= interrupt word
4	Allocate device	P1= device mnemonic P2= 0 CALL the service routine on interrupts. P2# 0 PON responses to the service routine. P2 is ignored for 'non-recognised' routines, to which responses will always be PONned.
5	De-allocate device	P1= device mnemonic
6	Clock interrupt	(all parameters ignored)
8	Ignored	
10	Execute chain	P1= address of caller's first TCB P2= sno for device returned from Allocate P3 is ignored P4= an IDENT field to be returned in P_P6 of TERMINATION and ATTENTION responses

5. Parameters and responses

DCU will only accept calls with DACT= 4, 5 or 10 (allocate, deallocate, execute chain) from "recognised routines". The acceptable service numbers (decimal) are 49, 50, 54, 57 and the routines are assumed to be TAPE, OPER, PRINTER, MK1FEADAPTOR respectively. These are the routines which are called to notify events which occur on devices allocated to routines with the appropriate service numbers (if direct calling rather than PONning was requested when the device was allocated). Reliable routines can be added to the list on request, by modifying the source and recompiling.

DCU will accept no calls before an initialise call (DACT=2). It does not check the SRCE number for DACT=2 (initialise), 6 (clock interrupt) or 3 (DCU interrupt). It could easily be made to do so.

When DACT= 2, 3 or 6, the value of the parameter on exit is undefined, and no reply will be PONned under any circumstances (although other routines may be called or messages PONned to them).

For DACT= 2 or 6, the value of the parameter on entry is immaterial. For DACT=3, P1 must have a copy of the DCU interrupt word.

For all calls on DCU, the reply will be found in P on exit (having overwritten the original value). For DACT = 4 (allocate), 5 (de-allocate) or 10 (execute chain), a copy of the reply will also be PONned to the caller. This is the only way that a response can get back to the caller if he PONned his original message to DCU instead of calling DCU. If he did call DCU, then the PONned reply is clearly redundant, and it can be suppressed by setting the most significant bit of SRCE before calling DCU. If that bit is set in a message PONned to DCU, then no reply will ever get back to the caller.

For other values of DACT, no reply is necessary. Nothing will be PONned to the caller, regardless of the setting of the top bit of SRCE. If DCU has been called, then P may have been overwritten, but its new contents are not specified here.

Values not specified in the responses from DCU are undefined. They need not be the same as the values supplied by the caller.

DACT=4, ALLOCATE P1 mnemonic

P2 =0: "call me to notify events" - only works for recognised routines
#0: "PON notification of events to me"

N.B. P2 does not affect the reply from the present allocate request; whether that is PONned or not depends on the top bit of SRCE. Nor does P2 affect the reply to any other call or PONned request to DCU. P2 controls how the user will be notified of interesting events which are not immediate responses to calls or PONned requests; for example, the termination of a transfer.

Response P1=1 unrecognised DACT no or mnemonic not found
2 device already allocated
3 caller not acceptable
0 O.K.

if P1=0: P2 device service no, needed for subsequent use of the device
P3 address of the device entry (q.v.)
P4 0, or -1 if device busy
P5
P6 mnemonic of device allocated. You need this to deallocate the line printer, if you asked for 'LP', because deallocate won't accept 'LP'.

But, for opers:
P3 address of buffer area (or 1's complement of no. of screens if device busy)
P4 no of screens
P5 size of buffer area

Note that P4<0 means that the device is still active: a response is expected, and execute chain requests will not be accepted until the response has arrived.

DACT=5, DE-ALLOCATE

P1 mnemonic

You may de-allocate a device which is allocated to someone else (DCU could check that easily), or a device which is busy.

Response

P1 = 0 O.K.
1 unrecognised DACT number
2 device not allocated
3 caller not acceptable
P2 = 1 mnemonic not found ('LP' will give this result, although allocate will accept 'LP')
P3 = address of device entry (why?)

DACT=8 IGNORED

On exit the parameter P will be undefined. No reply will be PONned under any circumstances.

DACT=10, EXECUTE CHAIN

P1 address of caller's first TCB
P2 service number for device (as returned by allocate)
P4 copied to the ID field in the device entry and returned to the caller with notification of termination of transfer.

The user must clear the response word to zero in each TCB before calling execute chain.

Response

P1 = 1 unrecognised DACT number or unrecognised service number
2 slot already on queue
3 source not acceptable
P3 = address of device entry

6. Action on stream response

The DCU routine response to an interrupt is to kick the appropriate service routine, which is called or ponned according to the P2 setting in the Allocate request. (Note that in the case of chains which terminate rapidly, for example faulty chains, the termination may be received before the DCU routine has returned to the caller, or ponned its acceptance of the request. In this case, if responses to Execute chain requests are to cause a call of the service routine, this call would be recursive, and the replies may be received in the 'wrong' order.)

In the activation of the service routine, the record P is set as follows:

(a) For ATTENTION responses

DACT = 3

P1: byte 2 is the attention byte

(b) For NORMAL TERMINATION responses

DACT = 2

P1 = address of last TCB executed in the chain

(c) For ABNORMAL TERMINATION responses

DACT = 5

P2 -1: timed out
0: see sense bytes
+1: DCU failure
8: sense bytes lost

P5 = address of last TCB executed in the chain.

Status information for the device is pointed to by the field SENSAD in the DEVICE ENTRY. It is laid out as follows: secondary status followed by tertiary status. If the relevant TCB had 'postamble valid' set, the sense bytes will be in the postamble; otherwise they will be in the appropriate fields in the device entry. When DCU hardware supports postambles, the software facility to put sense data in the device entry may be withdrawn.

If a chain has "timed-out", due to having taken longer than a value set at grope-time for the relevant device-type, then a special abnormal termination response, in which P2=-1, is sent to the service routine.

(d) For PCI responses

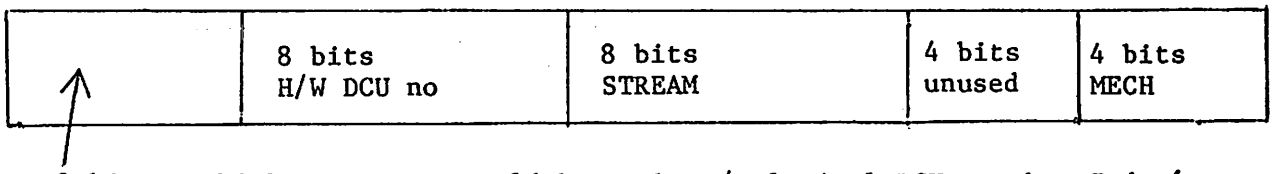
DACT = 6

P5 = address of a TCB which is later in the chain than the one whose termination caused PCI.

7. Major changes between GPC and DCU

1. DCU uses a pseudo comms area in which the first word has the image store address of the interrupt/activate words of the DCU. The rest of the area, if any, is used for work space for opers, as defined by the oper start and limit fields in the configuration table.

2. The word PTSM in the device slot is laid out like this:



8 bits - which I suppose could have the s/w logical DCU no, but I don't think this field is ever used in my code so it probably doesn't matter if you put something else in there.

3. The table PTS TO SLOT is replaced by a table DCU AND STRM TO SLOT which is laid out in order of logical dcu numbers, with 256 one-byte entries for the streams for each DCU.

4. DCU will not accept calls from anyone except the 'recognised service routines'. Other calls are rejected with P1=3. Users must agree on some segment for TCBs and the DCU must be initialised to use that.

5. DCU requires the caller to set up a chain of TCBs, instead of the RCB and logic block and address list of the GPC. The worst problem here is that each TCB must contain one word which is derived from the segment table entry for the segment containing the data. Also all TCBs must be in the same segment, so when you get a device allocated you are given an area (indicated by the fields UA SIZE and UA AD in the device entry) in which you must construct all your TCBs. Users must clear the response word to zero in each TCB before calling "execute chain".

J. Wexler

EMAS 2900 SUPERVISOR NOTE

No: 14
Date: 22/1/79

Tables Used by the Drum Driver

Two tables are set up by grope for routine DRUM.

The first is the DRUM CONTROLLER TABLE and is to be found at address COM_SFC TAD. There is one 6-word entry for each controller, corresponding to the following format:

```
recordformat CONTABF(integer ISCONTREG, BATCH, integername MARK, CRESP0)
```

where:

ISCONTREG is the image store address to send channel flags to the controller.

BATCH is a count of the streams for which activations have been set but not fixed.

MARK is a reference to the first word of the communication area.

CRESP0 is a reference to the controller response word for the SFC.

The second table is the drum table, containing one entry per drum. Different sizes of drum can be attached to the same SFC (!), so this table must contain all relevant information. Drums actually work with 1 Kilobyte sectors but the software makes them appear to work in Extended Sectors (ESECS), which are the same size as an EPAGE. If an integral number of ESECS does not fit on a track the remnant is discarded.

The format of a drum table entry is:

```
recordformat DTABF (integer NSECS, halfinteger LOGI, CONTI, integer c
                    SECLIM, NEXT, STATE, integername MARK, PAW, PIW, c
                    record ESCB0, ...., ESCBN (ESCBF))
```

where:

NSECS is the number of 1K sectors on the drum, ignoring any discarded because the track size is not an exact multiple of epage size.

LOGI is an index into an error logging table

CONTI is an index to the controller table for the controller of this drum.

SECLIM is the number of 1K sectors per track (again ignoring any discarded).

NEXT is the address of the next entry in the drum table (needed as this entry is of variable length).

STATE is the drum state - the sign bit set indicates an inoperable drum; otherwise a small integer.

MARK
PAW
PIW — these are references to the appropriate words in the communication area.

ESCB0 - ESCBN — extended sector control blocks - one for each extended sector; the number present = SECLIM/epage size. These control the queuing of transfers for each extended sector.

Each extended sector control block has the following format:

recordformat ESCBF(integer HQ, LQ, SAW0, PAWBITS, ADDSTRMS)

where:

HQ, LQ — are the heads of queues of high and low priority transfers. The sign bit is set when the top transfer is fired.

SAW0 — is the stream activation word for the first (1K) sector of this extended sector. Bit patterns for the remaining 1K sectors are generated from this one.

PAWBITS — A pattern of bits that, when ORed into the primary activation word, will start all 1K sectors of this extended sector transferring.

ADD STRMS — The address of the stream block (4 words: SAW0, SAW1, RESP0, RESP1) for the first 1K sector of this extended sector.

The queue of transfers pending is kept in normal parameter passing cells. The seven words are used as follows:

recordformat ESQF(integer RDEST, RSRCE, IDENT, ESECNO, STORI, ADPTS, c
longinteger LSAW, integer LINK)

where:

RDEST, RSRCE — are the destination and source for replying.

IDENT — is the returnable identifier provided with the request.

ESECNO — is the drum page - all drums being treated as one logical array of extended sectors.

STORI — is the index to the store array which identifies the core epage from which (or to which) the transfer is to take place.

ADPTS — = 0 for reads
= address of a count to be decremented on completion for writes.

LSAW — secondary activation word for first 1K drum sector of the extended sector.

LINK — to the PP cell containing the next transfer.

Notes

* The first 6 words are the same as the request, except that DEST and SRCE have been switched.

* The queues are circular - the header points to the last cell which points to the first. A pushdown LIFO arrangement will be tested shortly.

P.D. Stephens

Communications Record Format

The record format described below is extant from Chopsupe 186 onwards:

```
recordformat COMF(integer OCPTYPE, IPLDEV, SBLKS, SEPGS, NDISCS, c
        DDTADDR, GPCTABSIZE, GPCA, SFCTABSIZE, SFCA, SFCK, DIRSITE, c
        DCODEDA, SUPLVN, WASKLOKCORRECT, DATE0, DATE1, DATE2, c
        TIME0, TIME1, TIME2, EPAGESIZE, USERS, PROCMON, DQADDR, c
        SACPORT, OCPPORT, ITINT, CONTYPEA, GPCCONFA, FPCCONFA, SFCCONFA, c
        BLKADDR, DPTADDR, SMACS, TRANS, longinteger KMON, c
        integer DITADDR, SMACPOS, SUPVSN, PSTVA, SECSFRMN, SECSTOCD, c
        SYNC1DEST, SYNC2DEST, ASYNCDEST, MAXPROCS, INSPERSEC, ELAPHEAD, c
        COMMSRECA, STOREAAD, PROCAAD, SFCCTAD, DRUMTAD, SP0, SP1, SP2,c
        SP3, SP4, SP5, SP6, SP7, SP8, SP9, LSTL, LSTB, PSTL, PSTB, c
        HKEYS, HOOT, SIM, CLKX, CLKY, CLKZ, HBIT, SLAVEOFF, INHSSR; c
        SDR1, SDR2, SDR3, SDR4, SESR, HOFFBIT, S2, S3, S4, END)
```

This format describes "The Communication Record", which is kept locked in store at Public address X'80C00000'. It is readable at all ACR levels but writeable at ACR 1 only. Its purpose is to describe the hardware on which the EMAS 2900 System is running. Each entry is now described in more detail:

OCPTYPE The 2900 Processor on this configuration, as follows:
1 = 2950 or S1
2 = 2960 or P2
3 = 2970 or P3
4 = 2980 or P4
6 = 2976 or P4/1

IPLDEV The port/trunk/stream (or DCU/stream) of the device used at IPL time. (X'FFF' used for tape IPLS.)

SBLKS The no of 128K blocks of main store present.

SEPGS The no of extended pages for paging (i.e. not including any pages occupied by resident code & data).

NDISCS Then number of EDS drives available.

DDTADDR The address of the Disc Device Table which has NDISC entries, currently of 88 bytes each.

GPCTABSIZE The size in bytes of the GPC (or DCU) table.

GPCA The address of the GPC (or DCU) table.

SFCTABSIZE The size of the SFC (i.e. DRUM) table.

SFCA The address of the SFC table.

SFCK The number of (useable) 1K page frames of Drum store available for paging. (0 = No drum configuration)

DIRSITE — The Director site address (e.g. X'200') - no longer required?

DCOBEDA — The Disc Address of the Director (expressed as SUPLVN<<24!DIRSITE).

SUPLVN — The logical volume no of the disc from which the Sytem was "SLOADED". Various System components (e.g. DIRECT, VOLUMS) will page from here.

KLOKCORRECT — No longer used.

DATE0, DATE1, DATE2 — These three integers define the current date (updated at 2400) as a character string such that the length is in the least significant byte of DATE0.

TIME0, TIME1, TIME2 — These three integers define the clock time as a string in the same format as for DATE. The time is updated about every 2 seconds.

EPAGESIZE — The number of 1K pages combined to make up the logical "Extended Page" used in EMAS 2900. Currently=4.

USERS — The number of user processes (foreground+background) currently in existence. Includes DIRECT, VOLUMS and SPOOLR.

PROCMON — Was used for process monitoring. Now redundant.

DQADDR — The address of a public segment holding the header blocks for the disc transfer queues.

SACPORT — The bottom 16 bits hold the Port no of the Store Access Controller used to IPL the System. In dual SAC systems the other SAC is in the top 16 bits.

OCPPORT — The bottom 16 bits hold the Port no of the OCP used in the IPL. Like SACPORT, the top 16 bits will hold the Port of second OCP when duals are supported.

ITINT — The Interval Timer interval in microsecs. Varies between different members of the range.

CONTYPEA — The address of a 16 byte area containing the codes of the controllers in trunk order. Codes are:
0 = Not relevant to EMAS 2900
1 = SFC1
2 = FPC2
3 = GPC1

GPCCONFA, FPCCONFA, SFCCONFA — These three variables each point to a word array containing controller data. The first word in each case says how many controllers on the system. The remainder have Port&Trunk in top byte and Public segment no of comms segment in bottom byte. For GPCS the Public Seg no is apparently omitted!

BLKADDR — The address of first element of a word array bounds (1:SBLKS) containing the real address of each 128K block of main store. Real addresses are in the form RSN/SMAC NO/Address in SMAC.

DPTADDR — The address of the Disc Property Table. Now not needed as the DDT has addresses, not offsets, for properties.

SMACS — A bit mask of SMACS found at Grope time: 2**0 bit set if SMAC 0 found etc.

TRANS — The address of a 512 byte area containing 2 translate tables. The first is ISO to EBCDIC, the second the exact converse.

KMON — A 64 bit bitmask controlling monitoring of Kernel services. Bit 2**n means monitor service n. Bits can be set by Operator command KMON.

DITADDR — Disc index table address. The address of the first element of an array (1:NDISCS) containing the address of the disc device entries. Needed for S series and provided for compatablity on P series.

SMACPOS — The no of places that the SMAC no must be left shifted to be in the right position to access a SMAC image store location. Incredibly this varies between the 2980 and others!!

SUPVSN — The Supervisor identification number, as a three character string, e.g. 22A.

PSTVA — The virtual address of the Public Segment table which is itself a Public segment. All other information about PST can be found by looking at its own PST entry.

SECSFRMN — The no of seconds since midnight. Updated as for TIME.

SECSTOCD — The number of seconds to System closedown if positive If zero or negative no close down time has yet been notified. Updated as for TIME.

SYNC1DEST
SYNC2DEST
ASYNCDEST — These are the service nos N2,N3 & N4 for process parameter passing, as described in Supervisor Note 1.

MAXPROCS — The maximum number of paged processes that the Supervisor is configured to run. Also the size of the Process array.

INSPERSECS — The number of instructions the OCP executes in 1 second divided by 1000 (approx average for EMAS 2900).

ELAPHEAD — The head of a linked list of param cells holding service with an elapsed interval interrupt request outstanding.

COMMSRECA — The address of an area containing details of the Communication streams. (private to COMMS Control)

STOREAAD — The address of first element of the store record array bounds (0:SEPGS-1).

PROCAAD — The address of first element of the process record array bounds (0:MAXPROCS).

SFCCTAB, DRUMTAD — The addresses of two private tables provided by grope for use by the routine DRUM. They give details of the SFCS and DRUMS found on the system.

SP0->SP9 — Spare locations.

LSTL, LSTB, PSTL, PSTB, HKEYS, HOOT, SIM, CLKX, CLKY, CLKZ — These are the image store addresses for the following control registers:

Local Segment Table Limit & Base
Public Segment Table Limit & Base
Handkeys, Hooter, System Interrupt Mask Register
and the clock X,Y & Z Registers

HBIT — A bit pattern that when ORed into Control Register "HOOT" operates the Hooter. (0=Hooterless machine)

SLAVEOFF — A bit pattern (top 16 bits) and Image store address in bottom 16 bits. ORing the top 16 bits (after shifting) into the image store will stop all slaving of operands but not instructions.

INHSSR — A bit pattern and image location as for SLAVEOFF. ORing the bits into the location will switch off reporting of successful system retry.

SDR1, SDR2, SDR3, SDR4, SESR — The image store addresses of SMAC internal registers needed by the Engineers after SMAC errors have occurred.

HOFFBIT — A bit pattern that when ORed into a SMAC Engineers status register will stop reporting of errors from that SMAC.

P.D. Stephens

No: 16
Date: 21/11/79

GPC Device Entry Format

```
%RECORDFORMAT DEVICE ENTRY F(%INTEGER %C
     X1, %C
     GPTSM, %C
     PROP A, %C
     SECS SINCE, %C
     CA A, %C
     G RCB A, %C
     LB A, %C
     AL A, %C
     X2, %C
     RESP0, RESP1, %C
     SENSE1, SENSE2, SENSE3, SENSE4, %C
     X3, X4, %C
     IDENT, %C
     X5, %C
     MNEMONIC, %C
     DEVICE ENTRY S, %C
     PAW, %C
     U SAW 0, %C
     U RCB A, %C
     SENSE DATA A, %C
     LOG MASK, %C
     TR TABLE A, %C
     UA S, %C
     UA A, %C
     TIMEOUT, %C
     PROPS0, PROPS1)
```

where:

X1, X2, etc.	are spare.
GPTSM	GPC no, Port, Trunk, Stream, Mechanism.
PROP A	address of property codes obtained by GROPE.
SECS SINCE	time since current I/O request fired.
CA A	Communications Area address.
G RCB A	address of GPC's RCB.
LB A	
AL A	
RESP0, RESP1	response words set by GPC for most recent request. RESP1 = -1 after a time out.
SENSE1, 2, 3, 4	
IDENT	set by GPC from P4 of EXECUTE call.

MNEMONIC	device mnemonic.
DEVICE ENTRY S	size in bytes.
PAW	set by GPC from P3 of EXECUTE call.
U SAW 0	ditto.
U RCB A	set by GPC from P1 of EXECUTE call.
SENSE DATA A	address of sense data obtained by GPC after the most recent abnormal termination.
LOG MASK	used to control GPC's logging of abnormal terminations. Following an abnormal termination: 1. %UNLESS D_LOGMASK & byteinteger(addr(D_SENSE1)) = 0 %THEN 'dump the device entry to the main log'. 2. %UNLESS D_LOGMASK >> 8 = 0 %THEN 'read streams controller status'.
TR TABLE A	for an LP, the address of a translation table which turns characters which do not exist in the repertoire and which are not control characters into 'delete' (X'07'). This character is ignored by the printer. If the 'CART' on the printer is set to zero at initialisation time, translation has no effect i.e. f(i) = i.
UA S, UA A	the size in bytes and the address of a user area. Set up at GROPE time.
TIMEOUT	the time for which an I/O request is allowed to run before being failed 'timeout' by GPC.
PROPS0, 1	the device properties as obtained by GROPE.

A. Gibbons

No: 17
Date: 17/12/79

Basic Magnetic Tape Services

Services are offered on the relevant service number as supplied by Volumes.

Construct a record of the following format:

```
(integer DEST, SRCE, IDEN, halfinteger LENGTH, byteinteger c
 CONTROL, TYPE, integer ADR, LEN2, LSL, LSTBA)
```

where the fields have the following meanings:

DEST — X'310000' ! service number

SRCE — filled in by message-passing interface

IDEN — a private identifier

LENGTH — A signed number whose use is dependent on TYPE. If LENGTH=0 then the value in LEN2 is used in its place.

CONTROL — The individual bits may be interpreted as follows:

2**0 - Treat TM as block
2**1 - Suppress error retry
2**2 - Suppress short block checking
2**3 - Suppress long block checking

TYPE — Permissible values are:

0 - Erase one block
1 - read one block
2 - Write one block
4 - Check Read
5 - Private chain
8 - Skip a number of blocks
9 - Skip a number of tape marks
10 - Write a tape mark
17 - Rewind to BT

ADR — The address to be used in any data transfer.

LEN2 — May be used instead of LENGTH, especially where a size of greater than 32 Kbytes is required.

LSL — Local segment limit. This field and the next are only required if the user is specifying a private segment table to be used in the transfer of data (e.g. DOUT18). Note that this is always required where the data length exceeds 4 Kbytes. The bottom byte should be filled in with the user's ACR.

LSTBA — Local segment table base address.

The reply to a request is made in the following format:

(integer DEST, SRCE, IDEN, FLAG, P3, P4, P5, P6)

where the fields have the following meanings:

DEST — Originator of request

SRCE — request DEST field

IDEN — request IDEN field

FLAG — may have the following values:

0 - operation successful
1 - operation failed
2 - request rejected by TAPE routine
4 - unexpected TM, BT or ET, but otherwise successful

P3 — a value whose meaning is dependent on the request TYPE

P4 — Indicates when a short block or long block condition has arisen in an otherwise successful read request:

1 - short block
2 - long block

Details of action taken

ERASE (TYPE = 0)

A length of tape equivalent to a block of LENGTH bytes is erased.

READ (TYPE = 1)

A read to the given ADR is performed. The default response to failure is to attempt up to fifteen retries, reading in alternate directions.

If the length specified is negative then a backwards read is performed. The CONTROL bits may be used; these stop error retry and recovery from short or long block failures. If ADR is a local address then LSL and LSTBA are used to define a local segment table. The field P3 in the reply record will contain the number of bytes transferred, after a successful request.

WRITE (TYPE = 2)

A write from the given ADR is performed. The default response to error is to attempt up to five retries.

CHECK READ (TYPE = 4)

Similar to READ, but no data is transferred.

PRIVATE CHAIN (TYPE = 5)

See 'Private Chains', below.

FILE POSITION (TYPE = 8)

LENGTH blocks are skipped. If LENGTH is a negative number then the direction is backwards, otherwise forwards. Encountering a tape mark will normally cause a FLAG = 4. Note that the tape is positioned before the TM. Thus in normal mode it is impossible to exit a file using this command. However if CONTROL bit 2**0 is set then tape marks are treated as blocks. Field P3 in the reply is set equal to the number of blocks and tape marks skipped.

TAPE POSITION (TYPE = 9)

LENGTH x CONTROL tape marks are skipped, where conventionally; LENGTH specifies the number of files to be skipped, and CONTROL specifies the number of tape marks in each file.

A backwards skip is indicated by a negative LENGTH and may stop with FLAG = 4, indicating BT. A forwards skip will stop with FLAG = 4 either if ET is encountered or the first block in a file is a tape mark. In the second case the tape is left positioned between the tape marks.

Thus if CONTROL = 1 and LENGTH is large, the tape will be left at the first double tape mark found. Field P3 in the reply is set equal to the number of tape marks skipped.

WRITE TM (TYPE = 10)

One tape mark is written to the tape.

REWIND (TYPE = 17)

The tape is rewound to BT.

Queuing of Requests

Normally a user will wait for the reply to his current request before sending the next. However, some gain in speed can be made by sending a number of requests together. The first received will be actioned. The subsequent requests will be queued. As soon as the device terminates its first command another is released from the queue. Replies are sent normally and care may be required in the handling of errors, since execution of queued requests will continue regardless of the success or failure of prior commands.

Private Chains

This facility allows the user to specify his tape commands at a much more detailed level. The TAPE routine retains control of the deck, but passes the user's request directly to GPC. Private chain requests can be interspersed freely with the other commands documented here.

The user must provide:

a) A request control block (RCB)
b) A chain of logic block entries (LBEs)
c) A group of address list entries (ALEs)

The request format is:

DEST	X'310000' ! SNO
SRCE	0
IDEN	User-specified
P2	5
P3	Address of RCB
P4	A User service number if ATTNS are to be notified; otherwise 0
P5	0
P6	0

Information on RCBs, LBEs and ALEs should be sought in 'System Technical Description of GPC1' and the Engineer's manual for the tape deck concerned.

All data areas concerned, i.e. RCB, LBE, ALE and data buffers, must be 'locked down' before sending tbe request. This is achieved by use of DLOCK. They should be freed after the request has terminated by use of DUNLOCK.

Note that an ACR of 5 or lower is required to use these procedures.

The reply format is:

For Terminations:

P1	IDEN
P2	STREAM RESPONSE WORD 0
P3	(STREAM RESPONSE WORD 1<<14) ! Secondary Status Byte
P4 onwards	12 bytes of tertiary status information

For Attentions (usually obtained by use of DPOFF):

DEST	P4 from request (usually SYNC1 - i.e. UINFI(7))
P1	stream response word 0

Notes

1) The stream response word 0's have the SNO set in their top byte.

2) ATTNS will continue to be directed to the user until a non-private chain command is issued.

Modes of Operation

Eventually most interfaces by which a user can claim a tape will accept a MODE BYTE, to be passed on to the Supervisor tape routine. The format of this byte is:

Whole byte zero: 1600 BPI Phase Encoded (default)

bit 0 — 1 - interpret other bits

bit 3 — 0 - odd parity
1 - even parity

bit 5 — 1 - compress-expand (1900 mode)

bits 6&7

bit 6	bit 7	
1	1	1200 bpi (PE)
1	0	800 bpi (NRZI)
0	1	556 bpi (NRZI)
0	0	200 bpi (NRZI)

A. Anderson

TAPE VERSION 62 RELEASE NOTE

(ERCS04.TAPE62S, TAPE62Y) - MUST COMPILE PARMX.

CHANGES

1) NRZ 800 B.P.I. MODE SET IF 'CONTROL' TOP BIT IS SET. MODE RETURNS TO 1600 BPI P.E. WHEN TAPE IS RELEASED.

2) TAPE MARKS - COUNTED ON TAPE POSITION COMMANDS AND THE VALUE RETURNED IN REP_P3

3) CHECK READ - READ WITH NO DATA TRANSFER IMPLEMENTED (ERROR RECOVERY AS FOR NORMAL READ)

4) ANY SIZE OF LABEL NOW ACCEPTED. ONLY REQUIREMENT IS FOR EITHER:
VOL1 IN EBCIDIC FOLLOWED BY UP TO SIX EBCIDIC CHARACTERS.
OR SIX BYTES PRECEDING THE ABOVE THE FIRST FOUR OF WHICH ARE ZERO.
THIS COVERS IBM. AND AT LEAST THREE TYPES OF I.C.L. LABELS.

5) ROUTINE FREPORT BECOMES INTERNAL.

6) LABEL READING SPEEDED UP BY REDUCING FROM SIX STEPS TO FOUR

7) RELEASE/RETRIEVE CODE FOR V/MODE MAX = ON/OFF.
AWAIT VOLUMES & GPL CHANGES FOR TESTING.
ANTICIPATE REDUCTION OF TAPE SIZE BY 1½K.

TAPE ENTRY FORMAT (AS OF TAPE 64A)

This format is private to the supervisor tape routine and is found mapped onto space allocated in the GPC communications segments.

INTEGER(0) LINK - Address of the next entry. (Zero = end of list)

BYTE (4) SNO - Device service number.

BYTE (5) USE - The bits of this field define the current use of the deck.

1<<5 = READ LABEL SEQUENCE
1<<4 = UNLOADING
1<<3 = I/O REQUEST OUTSTANDING
1<<2 = I/O REQUEST VALIDATED
1<<1 = REQUEST EXTENDED

ie. A user request is first checked (USE = 4), then issued to GPC (use = 12), and if error recovery is required further commands are sent (use = 14).

BYTE (6) LAST REPLY - A copy of the P_P2 flag sent to the user in reply to his last request.

BYTE (7) USER COUNT - Set to one if the deck has been claimed by an ENTRY 12.

INTEGER (8) EXERRS - EXERRS is proportional to the number of operations performed on the deck since allocation. If it is exceeded by the number of faults then a warning is given. -EXCESSIVE ERROR RATE.

INTEGER (12) FAULTS - The number of failures on this deck since its allocation.

STRING(3) (16) MNEM - Deck mnemonic eg M03

INTEGER (20) LBE1 - Logic block entries. Up to four
LBE2 tape commands chained together.
LBE3
(32) LBE4

BYTE (36) NEXT - Variable controlling label reading sequence.

0 = NOT IN PROCESS
1 = SEND PROPERTIES + SENSE
2 = REWIND
3 = READ BLOCK
4 = REWIND (IF UNLABELLED)

BYTE (37) PROP1 - Properties codes returned by
PROP2 SEND PROPERTIES command.
(39) PROP3

INTEGER (40)	LOAD	-	Status of deck

-2 = INOPERABLE
0 = NOT LOADED
1 = LOADED WITH LABEL
2 = LOADED NO LABEL

INTEGER (44)	STREAMR0	-	Stream response words.
(48)	STREAMR1		These are the 'reply' from the hardware to the last command. (task byte set = SNO)

INTEGER (52)	OPS	-	Number of READ/WRITE operations performed since allocation.

INTEGER (56)	ALE1	-	Address list entries.
	ALE2		These are two descriptors for
	ALE3		use in defining data for the
(68)	ALE4		LBE's.

BYTE (72)	SSTAT	-	Secondary status byte
BYTE (73)	T0	-	Tertiary status bytes.
	T1		
	T2		
	T3		
	T4		
	T5		
	T6		
	T7		
	T8		
	T9		
	T10		
(84)	T11		

RECORD REQ (88) . Details of current user request.

INTEGER (88)	DEST		
	SRCE		
	IDEN		
(100)	LENGTH	-	size of data
(104)	SPAN	-	address of data
BYTE (108)	CONTROL	-	Bits amend command (see SUP 17)
	TYPE	-	command
	DUM1		
	FAULTS	-	faults on this request
INTEGER (112)	LSL	-	Local segment limit
	LSTBA	-	Local segment base address
	COUNT1	-	Used for counting blocks & TM's.
(124)	COUNT2		"

RECORD RCB (128) - Request control block

INTEGER (128)	LIMFLAGS	-	see GPC spec.
	LSTBA		
	LBBYTES		
	LBA		
	ALBYTES		
	ALA		
(152)	INITWORD		
INTEGER (156)	LAB1	-	Space used for label reads.
	LAB2		
STRING (7) (164)	DSN	-	Label

BYTE (172) ORGDIR - Original direction of movement.
1 = FORWARD
0 = BACKWARD

BYTE (173) UNLOADPENDING - IF SET then unload deck when current command terminates.

BYTE (174) ORGREQ - Original command
Note/ REQ_TYPE is overwritten during error retry etc.

BYTE (175) BITS Defines mode and B.P.I.

INTEGER (176) FIRSTPEND - Start of queue of pending requests.
LASTPEND End " " "

Further information on several of these fields can be found in the GPC descriptions and in Supervisor note 17.

Alan Anderson 21/4/80

Table of Service numbers for MAXPROCS = 128

Proc	SYNC1	SYNC2	ASYNC	Proc	SYNC1	SYNC2	ASYNC
1	X0C1	X141	X1C1	65	X101	X181	X201
2	X0C2	X142	X1C2	66	X102	X182	X202
3	X0C3	X143	X1C3	67	X103	X183	X203
4	X0C4	X144	X1C4	68	X104	X184	X204
5	X0C5	X145	X1C5	69	X105	X185	X205
6	X0C6	X146	X1C6	70	X106	X186	X206
7	X0C7	X147	X1C7	71	X107	X187	X207
8	X0C8	X148	X1C8	72	X108	X188	X208
9	X0C9	X149	X1C9	73	X109	X189	X209
10	X0CA	X14A	X1CA	74	X10A	X18A	X20A
11	X0CB	X14B	X1CB	75	X10B	X18B	X20B
12	X0CC	X14C	X1CC	76	X10C	X18C	X20C
13	X0CD	X14D	X1CD	77	X10D	X18D	X20D
14	X0CE	X14E	X1CE	78	X10E	X18E	X20E
15	X0CF	X14F	X1CF	79	X10F	X18F	X20F
16	X0D0	X150	X1D0	80	X110	X190	X210
17	X0D1	X151	X1D1	81	X111	X191	X211
18	X0D2	X152	X1D2	82	X112	X192	X212
19	X0D3	X153	X1D3	83	X113	X193	X213
20	X0D4	X154	X1D4	84	X114	X194	X214
21	X0D5	X155	X1D5	85	X115	X195	X215
22	X0D6	X156	X1D6	86	X116	X196	X216
23	X0D7	X157	X1D7	87	X117	X197	X217
24	X0D8	X158	X1D8	88	X118	X198	X218
25	X0D9	X159	X1D9	89	X119	X199	X219
26	X0DA	X15A	X1DA	90	X11A	X19A	X21A
27	X0DB	X15B	X1DB	91	X11B	X19B	X21B
28	X0DC	X15C	X1DC	92	X11C	X19C	X21C
29	X0DD	X15D	X1DD	93	X11D	X19D	X21D
30	X0DE	X15E	X1DE	94	X11E	X19E	X21E
31	X0DF	X15F	X1DF	95	X11F	X19F	X21F
32	X0E0	X160	X1E0	96	X120	X1A0	X220
33	X0E1	X161	X1E1	97	X121	X1A1	X221
34	X0E2	X162	X1E2	98	X122	X1A2	X222
35	X0E3	X163	X1E3	99	X123	X1A3	X223
36	X0E4	X164	X1E4	100	X124	X1A4	X224
37	X0E5	X165	X1E5	101	X125	X1A5	X225
38	X0E6	X166	X1E6	102	X126	X1A6	X226
39	X0E7	X167	X1E7	103	X127	X1A7	X227
40	X0E8	X168	X1E8	104	X128	X1A8	X228
41	X0E9	X169	X1E9	105	X129	X1A9	X229
42	X0EA	X16A	X1EA	106	X12A	X1AA	X22A
43	X0EB	X16B	X1EB	107	X12B	X1AB	X22B
44	X0EC	X16C	X1EC	108	X12C	X1AC	X22C
45	X0ED	X16D	X1ED	109	X12D	X1AD	X22D
46	X0EE	X16E	X1EE	110	X12E	X1AE	X22E
47	X0EF	X16F	X1EF	111	X12F	X1AF	X22F
48	X0F0	X170	X1F0	112	X130	X1B0	X230
49	X0F1	X171	X1F1	113	X131	X1B1	X231
50	X0F2	X172	X1F2	114	X132	X1B2	X232
51	X0F3	X173	X1F3	115	X133	X1B3	X233
52	X0F4	X174	X1F4	116	X134	X1B4	X234
53	X0F5	X175	X1F5	117	X135	X1B5	X235
54	X0F6	X176	X1F6	118	X136	X1B6	X236
55	X0F7	X177	X1F7	119	X137	X1B7	X237
56	X0F8	X178	X1F8	120	X138	X1B8	X238
57	X0F9	X179	X1F9	121	X139	X1B9	X239
58	X0FA	X17A	X1FA	122	X13A	X1BA	X23A
59	X0FB	X17B	X1FB	123	X13B	X1BB	X23B
60	X0FC	X17C	X1FC	124	X13C	X1BC	X23C
61	X0FD	X17D	X1FD	125	X13D	X1BD	X23D
62	X0FE	X17E	X1FE	126	X13E	X1BE	X23E
63	X0FF	X17F	X1FF	127	X13F	X1BF	X23F
64	X100	X180	X200	128	X140	X1C0	X240

Roderick McLeod